(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 October 2002 (31.10.2002)

PCT

(10) International Publication Number
WO 02/086684 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/GB02/01856

(22) International Filing Date: 22 April 2002 (22.04.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0110131.0 24 April 2001 (24.04.2001) GB
0127735.9 20 November 2001 (20.11.2001) GB

(71) Applicant *(for all designated States except US)*: **HEWLETT-PACKARD COMPANY** [US/US]; 3000 Hanover Street, Palo Alto, CA 94304 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **PROUDLER, Graeme, John** [GB/GB]; 5 Touchstone Avenue, Stoke Gifford, Bristol BS34 8XQ (GB).

(74) Agent: **LAWRENCE, Richard, Anthony**; Hewlett Packard Limited, Intellectual Property Section, Filton Road, Stoke Gifford, Bristol BS12 8QZ (GB).

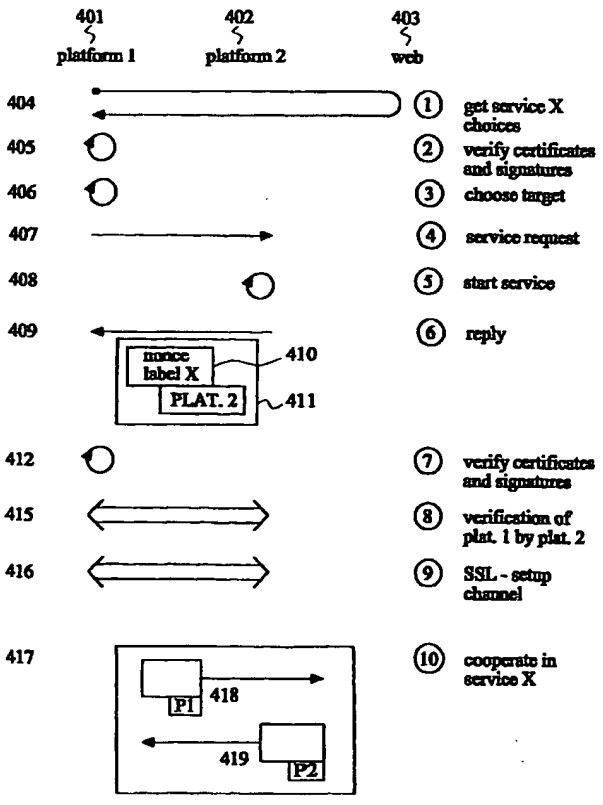
(81) Designated States *(national)*: JP, US.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: AN INFORMATION SECURITY SYSTEM



(57) Abstract: An information security system is disclosed having a considerably simplified access control infrastructure. The number of secrets in a computer system domain is reduced to a minimum, yet individual users may still be identified and access to applications may still be individually controlled. The trusted entity in each of a plurality of platforms (100, 200, 202, 203) of the computer system may store an identity secret of the platform (100, 200, 202, 203) and may be trusted to use that secret in conjunction with an information label only when the platform (100, 200, 202, 203) is running the correct software to provide and/or take part in a particular service associated with that information label.

WO 02/086684 A2

An information security system

This invention concerns an information security system, primarily but not exclusively for use in commercial domains.

The invention concerns the simplification of information security systems, to reduce the cost and complexity to the domain in using the security service.

Background

One reason why commercial enterprises are reluctant to embrace information security systems is that they require a complex on-line infrastructure, which is relatively difficult and expensive to maintain. A second reason is that applications that use security must be individually written to take advantage of security. This makes them more expensive to buy and costly to maintain. A third reason is that it is relatively simple for an operator to make a mistake and compromise security mechanisms.

Those skilled in the art of information security will be aware of the use of secrets as authorization and authentication information. Possession of a secret is taken as proof of the right to use or provide a service. In some systems, some secrets (such as passwords) are confidentially communicated between the user of a service and a provider of a service. Preferably, however, possession of a secret is proven without disclosing that secret. There are several such methods, generally involving the use of the secret as an input to an algorithm whose inputs are statistically impossible to

deduce from its output. Challenge-and-response protocols involving hash algorithms (such as SHA-1) and symmetric or asymmetric cryptographic algorithms (such as 3DES or RSA) may be used for such purposes.

Such systems require the distribution of secrets between the parties. The distribution of secrets such as passwords and symmetric keys requires the use of channels that maintain the confidentiality of those secrets, e.g. a secure socket layer (SSL) protocol. There are numerous such distribution systems. Such secrets may be distributed using confidential channels provided by existing secrets (leading to a hierarchy of secrets). Such secrets may also be distributed through confidential "out-of-band" channels. Those skilled in the art of information security will be also aware of Public Key Infrastructures (PKI). Such systems are trust-based methods that distribute the public keys of asymmetric cryptographic algorithms. Preferably, one entity generates an asymmetric key pair and keeps secret the private key of that pair. The entity then uses a PKI to distribute the public key of that key pair. The PKI issues "digital certificates", which contain a statement of the public key of an entity and the label of that entity, all signed by the (secret) private key of some Certification Authority (CA). The public key of the CA is well publicized. So, when a third party receives a certificate, it can use the CA's public key to verify the contents of the certificate. When a certificate has been verified by such a method, the third party accepts that the stated public key belongs to the stated entity, because the third party trusts the CA. The third party can therefore use the public key from the certificate to

verify the authenticity and integrity of data signed by the stated entity.

Some static PKI infrastructure is usually necessary, in order to identify entities. The complexity of the infrastructure rapidly increases, however, when multiple secrets have to be distributed and stored for each user/application pair, and especially when secrets have to be dynamically verified because each platform cannot store all access control information.

Existing security systems tend to have some security functions in the platform and some in individual applications. A platform might have shared security resources, such as a cryptographic accelerator, or a cryptographic library, or a store for secrets, or an SSL engine, for example. A security aware application might use those shared resources when performing its security functions. Those security functions include provision of confidentiality, integrity, authentication, non-repudiation, and so on. Such shared services and individual functions are well known to those skilled in the art of information security. Often each application requires its own security secrets. Duplication of security services is in conflict with preferred security practice, where the number of mechanisms that deal with security should be as small as possible and as tested as possible, to minimize the risk of bad design or poor construction.

The management of access to a service becomes increasingly difficult with increasing complexity and size of service. Each customer (user) is provided with authorization data to permit access to such a service. Methods of user

authentication are well known to those skilled in the art. Such user authentication data may take the form of a simple password, biometric measurements, or (preferably) cryptographic keys, for example. Each computer platform providing the service must be capable of recognizing such authorized users, in order that the service is restricted to bona fide users. Each such computer platform must also be provided with authorization data that permits users to validate the service, in order that users may be convinced that a service is bona fide. Each such computer platform providing the service must also be provided with authorization data that permits each such computer platform to identify and validate each other such computer platform. This is necessary in order that a service can be distributed over multiple bona fide computer platforms. Such distribution is a commercial advantage because it enables greater reliability and more economically efficient load distribution. Clearly, a great deal of authorization information can be required. The need to distribute and manage this on-line authorization information is a significant drawback of existing methods of identification and verification at entry points to an electronic service. One such system is the well-known "Kerberos" system [see, for example: "Kerberos: an authentication service for computer networks", Neuman and T'so, IEEE Comms, September 1994]. This is a gatekeeper mechanism, used to verify access rights based upon long term secrets, and to distribute temporary secrets that provide short term access to resources.

Neither the commercial customer nor the commercial domain always benefit from the conventional security model described above; using secrets to identify a user,

registering with the domain, obtaining a token to use the computing engine or service, and submitting the token in order to actually use the resource. The extra complexity costs both time and money, both of which are disadvantageous.

Those skilled in the art of information security will be also aware of the Trusted Computing Platform Alliance (TCPA) (details of which can be found at www.trustedpc.org and also in WO 00/48063, the contents of which are incorporated herein by reference). This industry body has defined the concept of a Trusted Computing Platform. Essentially, two roots-of-trust are built into each platform. One is the root-of-trust-for measurement, which starts the process of measuring the software environment in the platform. The second is the root-of-trust-for-reporting, which stores and reports summaries of the measurements made by the root-of-trust-for measurement. The root-of-trust-for-reporting is usually called the Trusted Platform Module (TPM), because it is typically implemented as a single integrated circuit. The TPM protects its methods from interference. The TPM protects its secrets from observation and interference. The TPM contains several cryptographic functions, such as random number generation, key generation, encryption, and decryption. By means of a TCPA protocol, a TPM may obtain multiple anonymous trusted cryptographic identities. A CA chosen by the owner of that TPM grants each such trusted identity. Each such trusted identity can be used to cryptographically prove that certain data came from a trustworthy computing engine (the TPM). The TCPA specification referred to above also describes how an integrity response may be evoked from a platform. An

integrity response provides evidence of the software state of that platform by reporting (amongst other things) the summaries of the measurements made by the root-of-trust-for measurement. An integrity response signed by a trusted identity is sufficient to prove that the integrity response from a platform can be believed (the private key belonging to the trusted identity is used to sign the integrity response from the platform containing that identity). The recipient of the integrity response must compare the response with a set of measurement values that should be obtained if the platform is in a trusted state. Methods of obtaining such comparative values are described in the TCPA specification referred to above, and will not be described here. It should be noted, however, that a signed statement of the summaries of the measurements made by the root-of-trust-for-measurement in a correctly operating platform is a convenient method to permit simple verification of an integrity response. The TCPA system relies upon a PKI; the system relies upon several certificates issued by several entities, including the manufacturer of the TPM, the manufacturer of the platform, a conformance laboratory [such as one that deals with conformance to the international "Common Criteria" security description], and the manufacturers of components of the platform (including the software in the platform). The TCPA specification discloses a method of measurement of the software state of a platform and the summarizing of that state as a number of Platform Configuration Registers (PCRs) inside a Trusted Platform Module, which is much better physically protected against interference and prying than is the rest of the platform. The TCPA specification describes a pair of commands called TPM_SEAL and TPM_UNSEAL. A TPM that receives the command TPM_SEAL

(plaintext, pcr-index, pcr-value) encrypts the plaintext data along with the pcr-index and pcr-value. A TPM that receives the command TPM_UNSEAL(ciphertext) internally decrypts the ciphertext data to obtain the data [plaintext, pcr-index, pcr-value]. At this stage, the plaintext data is hidden inside the TPM. The TPM exports the plaintext data out of the TPM only if the current pcr-value of pcr-index in the platform matches the decrypted value of [pcr-index, pcr-value]. Hence data is revealed by the TPM only if the platform is currently in the state that was stated when the plaintext data was encrypted.

Existing security services use conventional methods to control and provide security services. Secrets are distributed and loaded using key distribution systems and the facilities of a PKI. Each platform providing a service has its own secrets, and other such platforms use those secrets to identify and verify platforms that provide the service. Each user of a service is provided with secrets that enable it to access the service. Each platform providing a service needs proof, either individually or via another platform, that a prospective user actually has the secrets that prove the right to access the service. When TCPA Trusted Platforms have been deployed, existing services may be amended so that: (1) platforms may be identified by trusted identities, (2) platforms may verify each other's integrity before providing the service, using the service, or cooperating to provide the service.

A previous patent application (Trusted States) PCT/GB00/03613, the contents of which are incorporated herein by reference describes how a platform may exist in a variety of different states, each state optimised to

ensure the validity of some electronic service. The integrity response provided by the platform serves to prove to a prospective user that the platform is in a particular state that is suitable to safely provide that service. Those skilled in the art of providing an electronic service will be aware that a service may be provided by a single platform or may be provided by one platform cooperating with at least one other platform, and different services may execute upon the same platform. A previous patent application (Performing a service on a computer) GB 00 20441.2, also incorporated herein by reference, describes the use of enhanced compartments to provide specification and isolation and audit of such services.

Existing security systems follow the military model, where users and computing engines have rights and privileges to perform certain actions and use certain data. There are disadvantages in that such a model is not always necessary for a commercial domain wishing to provide a computing resource. The domain might be providing a conventional electronic service, or might be providing just a computing engine upon which a user can execute his own data and applications. Indeed, the domain needs to provide such an engine service for itself, in order to partition its platforms as a set of arbitrary computing engines. An important concern of many such domains is that a user will pay for the resources consumed. This may be true even when the user belongs to the domain, because corporate accounting models often charge individual projects or individual departments for use of corporate resources. Of course, it is still possible that a domain needs just to confirm that the "customer" or user has the right to use a

Reception by the platform of a cryptographic challenge incorporating one of said at least one labels from a second platform may cause the platform to determine whether the computing resources and/or software associated with said label can be provided by the platform.

Proof of possession of a label by a platform may be sufficient for another entity to cooperate with that platform for the purposes of using and/or providing the computing resources and/or software described by that label.

The computer system may be operable such that the right to use the computing resources and/or software described by the label depends on provision of one or more of:

proof of possession of a platform secret,
proof of possession of a user secret,
presentation of a non-secret authorisation value associated with a user whose use is known to be indicative of a request from the user,
presentation of a non-secret authorisation value associated with a user whose use is known to be indicative of agreement by the user to tender payment.

At least one platform may contain trustworthy integrated mandatory enforcement controls and security capabilities that transparently provide security and privacy to applications that are at least substantially ignorant of security and privacy, and preferably requires permission from at least one other platform to permit the flow of information to the resources allocated to said other

confidence in the results obtained from arbitrary platforms.

It follows that there is an advantage in reducing the complexity of a traditional security system, even if the absolute level of security is less than that provided by a conventional security system. The basic platform should provide security, in order that applications can be ignorant of security. There should be rapid access to computing resources and services, without having to use gatekeeper authorization mechanisms.

The invention

According to the invention a computer system comprises at least one platform containing a trusted entity and at least one label, the trusted entity being operable such that use of the or each label by the trusted entity is dependent on the presence or potential presence of a predetermined software state in the or each platform.

The at least one label may be adapted to indicate or advertise the presence or potential presence of the predetermined software state in the or each platform.

The presence of the predetermined software state may be an indication that the trusted entity is capable of providing a particular computing resource or service. The potential presence of the predetermined software state may be an indication that the trusted entity is capable of providing a particular computing resource or service.

The or each label may describe a service which can potentially be offered by the at least one platform.

The computer system advantageously provides a trusted apparatus which is operable to indicate when a particular computing resource or service may be available from a trusted entity.

The predetermined software state may include a particular configuration of computing resources and/or software described directly or indirectly by the or each label.

Labels in at least two platforms may be the same where the labels describe essentially the same configuration of computing resources and/or software. The labels in the two platforms may be essentially the same where the labels describe a particular configuration of computing resources and/or software related to the same distributed computing engine or distributed service.

The or each label may be widely published and one form of published label may be signed using a secret known to the platform. One form of published label may include descriptive information and is signed by a trusted entity. One form of published label may include descriptive information about the configuration of computing resources and/or software associated with the label and is signed by a trusted entity. One form of published label may include an offer to provide a configuration of computing resources and/or software associated with the label. The or each label may be signed using a secret known to the platform.

service and/or resource, and that no payment is required. This conventional security model is still appropriate for an employee of a company or a citizen of a country whose ID permits access to certain resources, as a matter of right.

A given user wanting a computing resource may wish to use applications and data provided by that resource, or may wish to obtain access to a virgin computing engine, or a computing engine executing some limited amount of software such as an operating system but few or no applications. The user may partition his processing requirements according to level of security/privacy, and distribute a task amongst domains according to those security/privacy criteria. Non-sensitive computing threads may be executed on arbitrary platforms such as individual workstations connected to a corporate network or to the Internet, rather than executed by a Data Center. A user (or even a domain) could take advantage of unused computing resources in one time zone when computing resources in another time zone are stretched. A private individual could execute sensitive data on his own platform and have a reciprocal agreement to execute insensitive data on an arbitrary platform belonging to a private individual who is asleep, and whose platform is idle. A roaming private individual could execute sensitive data on his own Personal Digital Appliance (PDA) and have an agreement to execute less sensitive data on an arbitrary platform in the logically local environment. A multinational corporation could use its worldwide desktop workstation resources for corporate processing when employees are not at their desks. The same non-sensitive thread could be executed on more than one platform, and the results compared, to provide increased

platform from the resources allocated to the first-mentioned platform.

According to another aspect of the invention a computer system comprises at least one platform containing a trusted entity and at least one label, the trusted entity being operable such that use of the or each label by the trusted entity is dependent on the presence or potential presence of a predetermined software state in the or each platform, wherein the at least one label is adapted to indicate or advertise the presence or potential presence of the predetermined software state in the or each platform, and wherein the or each label is widely published and describes a service or resource which can potentially be offered by the at least one platform.

According to a further aspect of the invention a computer system comprises at least one platform containing a trusted entity and at least one label, wherein the label describes a predetermined software state in the or each platform and wherein the trusted entity is operable to use the label if the predetermined software state is described by the label is present or potentially present in the or each platform.

The trusted entity may sign the at least one label with a secret known to the platform only if the predetermined software state is present or potentially present in the at least one platform.

The at least one label may publicly disclose the predetermined software state in order to indicate the

availability of a service or the resource on the or each platform.

According to the invention a computer system comprises at least one platform containing a trusted entity and at least one application, wherein the platform is operable to perform security functions for the computer system.

The platform preferably performs substantially all security functions and the applications preferably perform substantially no security functions.

The platform may be operable to apply mandatory security controls on communications from the computer system. Updates of security functions may be broadcast across the system to the at least one platform.

According to the invention a method for a computer system to signal the potential availability of a computing resource comprises providing a platform containing a trusted entity with at least one label, wherein the label is used by the platform only when a predetermined software state is present in the platform.

The label may describe the computing resource or service, which is defined by the predetermined software state.

An information security system may have a level of overall security slightly less than that of a conventional security system, depending on exact circumstances, but it is anticipated that the level of security is adequate for commercial purposes.

One aspect of the invention concerns the simplification of access control infrastructure that must be provided by the domain. The number of secrets in the domain is reduced to a minimum, yet individual users may still be identified and access to applications may still be individually controlled. At the same time, secure applications may be maintained across the domain by a broadcast mechanism, without having to deal with each platform as an individual. A mechanism of payment for resources is an integral part of the system. The requirement for a domain's on-line Public Key Infrastructure (PKI) is reduced, or even eliminated.

Another aspect is the transfer of all security mechanisms into the platform, away from applications. This simplifies applications and makes them less of a risk, because they are no longer required to properly implement security functions. This also makes it possible to apply security to legacy applications that are not aware of security. This also reduces risks, because the platform can provide mandatory security controls and because there is a reduction in the number of people required to design and build security mechanisms and controls.

This invention also recognizes that a major requirement of a user of such domains is that a user's data and applications on a computing engine do not "leak" to other users. If a user chooses to cache applications and information with a domain (for whatever length of time), the domain needs to provide access controls to those applications and data, such that only the user or his agent can use the applications and data.

This invention discloses an architecture that potentially eliminates the need for applications to be aware of security; minimizes the number of secrets in the platform (just one secret per platform); minimizes the number of secrets that identify the user (just one secret/value per user); permits all secrets to be installed at manufacture or initialisation, yet enables separate access controls for separate users to separate services or domains; enables the broadcast of maintenance information over all platforms in the domain, irrespective of platform identity; and does not necessarily need an on-line PKI infrastructure.

One main aspect of the invention is that a trusted entity in each of a plurality of platforms stores an identity secret of the platform and can be trusted to use that secret in conjunction with a label only when the platform is running the correct software to provide and/or take part in a particular service associated with that label.

A second aspect of the invention is the publication of global labels describing services and computing engines. Each global label preferably serves to identify the properties of an engine and/or service executing on that engine, and is preferably signed by a platform hosting the engine/service. Such labels and their interpretations may be published on a database, or on the World Wide Web, and may be available within a domain or globally. Such labels and their interpretations are preferably digitally signed by a trusted entity, preferably the domain itself. A platform preferably publishes the global labels of the engines and/or services that it has the potential to

provide, even if the platform is not currently providing or in a position to provide that engine and/or service.

A third aspect of the invention is that a platform determines whether an engine and/or a service associated with a label can actually be provided only when a platform is requested to provide a particular engine and/or service associated with a label.

A fourth aspect of the invention is that a platform may provide and/or use an engine and/or a service associated with another platform based solely upon recognition of a label owned by that other platform.

A fifth main aspect of the invention is that a domain need not provide its own on-line PKI to validate users using secrets. Instead, the domain may either use an external on-line service that may use true secrets or may use quasi-secrets. Preferably the on-line service is one that can be used for user identification and more preferably the on-line PKI is one that can be used for on-line credit verification and payment.

A sixth aspect of the invention is that each of a plurality of platforms has an integral security service that provides security/privacy functions that drastically reduce (if not eliminate) the level of security functions that individual applications on the platforms must incorporate. The security service preferably enables an operator and/or a user to set the policy governing the level of security and privacy that will be provided for the engines and/or services that execute on the platform.

Each platform may be considered to have one or more cryptographic identities that are published only if the platform is in a certain state. Each cryptographic identity preferably consists of a different global label and the same shared secret for use in cryptographic algorithms. The identity and the mechanism that publishes global labels are preferably trusted, so that a third party believes that a platform is in the appropriate state to provide a particular engine and/or service when the platform exports that identity. Each such state preferably corresponds to an environment where mandatory controls are applied to a user's data and applications. These controls preferably isolate the user's data and applications as and when necessary. Such states may also correspond to services executing in such an environment.

All identities publicizing the same engine and/or service preferably have the same label, across all platforms, at least within the domain. This permits easy identification of platforms that can cooperate to provide those engines and/or services. A label may however be differentiated to indicate that a platform is executing the client portion of a client-server process, the server portion of a client-server process, or a peer portion of a peer-peer process, for example. As mentioned previously, all identities belonging to the same platform preferably use the same cryptographic secret, but have different labels. If there are privacy concerns, some or all identities belonging to the same platform could have different cryptographic secrets, but this increases complexity. Identities belonging to different platforms and/or systems must preferably have cryptographic secrets that are (statistically) different. Identities could be obtained

via the TCPA protocol for obtaining platform identities, but are preferably preloaded or maintained directly by the domain itself.

Each identity secret preferably has an associated credential in the form of a digital certificate. That credential preferably attests to the name and/or nature of the platform containing the secret. That credential is preferably digitally signed by an entity that is considered trustworthy by third parties that wish to interact with the platform containing the identity secret. Preferably that trusted entity is the domain itself.

Each label preferably has an associated credential in the form of a digital certificate. That credential preferably attests to the name and/or nature of the engine and/or service described by that label. That credential is preferably digitally signed by an entity that is considered trustworthy by third parties that wish to interact with platforms providing that engine and/or secret. Preferably that trusted entity is the domain itself.

When a platform asserts that it can provide an engine and/or service, it preferably uses its identity secret to sign advertising data comprising at least the label describing that engine and/or service and an indication that the data is an advert. A platform posts or retracts appropriate adverts to/from a web site (for example), preferably according to whether it has the potential to provide the engine and/or services that are associated with that label. Each advert is preferably associated with an address (such as an IP address), to enable

communications with the platform. The existence of such advertising does not imply that the platform will provide that service; only that it has the potential to provide that service. On receipt of a request, a platform preferably assesses its internal state, which has been measured in a reliable manner, preferably using TCPA mechanisms. The platform (strictly, the platform's TPM) preferably uses its identity secret to sign confirmation of the availability of an engine and/or service only if the platform's state is the correct state to provide the engine and/or service. Such confirmation data preferably includes at least the label describing that engine and/or service and an indication that the data is a confirmation.

Preferably both the user and the host use platforms whose architecture complies with this invention. Otherwise, one platform can provide or consume the service, but two platforms cannot cooperate, and/or cooperate to provide a distributed service to a third platform, in the manner envisaged in this invention.

If a user wishes to use a particular engine/service, the user or his agent preferably browses Web sites and searches for adverts of that particular type of engine and/or service. When the user finds a suitable label, it verifies the associated credentials and decides whether to trust the attestation. If the user trusts the attestation sufficiently for the task in hand, he preferably contacts the indicated platform in the indicated domain, states the target engine/service, presents any payment information or identification information that may be necessary, and presents any relevant policy information. A domain platform may decide whether or not to grant access based

upon the user's credentials. Alternatively, the domain may grant access irrespective of the user's credentials.

After two platforms have agreed to cooperate, they preferably use conventional cryptographic methods such as the Diffie-Hellman (DH) protocol or the Secure Socket Layer (SSL) to provide confidentiality of communication between them. So the long-term identity secret inside the platform provides identification of the platform, while less permanent secrets created by the DH protocol or SSL, preferably provide confidentiality for the platform and/or for the engines and services provided by the platform. The user preferably sends to the domain a policy statement that states the security and privacy conditions that apply to the data and applications that will be executed by the domain. Such a policy will often indicate just that defaults are to be used. Such defaults are preferably that that all user data and applications are to be isolated from that of other users while they are on the host. The user may also intend to cache certain data and applications at the domain's platform. Then the host must isolate and store and protect such data and applications, such that only the user can access them, based upon the user's identity or the identity of the user's platform. These protection mechanisms do not require the involvement of the user. The domain may have trusted facilities that provide the necessary protection, probably but not necessarily involving cryptographic processes and the storage of secrets known only to the domain and its agents. The default policy may be that the user can gain access to those data and applications if the user's platform has the correct identity-label and the user presents the same generic payment/access information.

In one preferred method of use, domains are not aware of the identity of the actual user or users. Platforms constituting a distributed service or distributed computing engine in a domain will preferably cooperate with other platforms having a global label describing that distributed service or computing engine and signed by the domain. This model is particularly applicable to a group of platforms configured to belong to a particular set, such as a computing center for a domain.

In another preferred method of use, domains may require the presence of a user or users at a platform, or at a particular platform. In that case, the user's platform may pass the user's credentials to the domain. Preferably, however, the platform may export a particular label only when a certain user or users are present, it being the duty of the platform to verify that the user is actually present. The platform may use any of the conventional recognition methods to verify that a user is present, including passwords, biometrics, location, and security tokens such as smartcards. This model is particularly applicable when a user has a PDA and wishes to use computing resources in the domain. In one sense, the user's platform becomes the user's token as far as the network is concerned.

A platform requires a trusted mechanism to control the use of the identity secret and the global labels. The identity secret is preferably used by the TPM to sign a given label only if the platform is in the state that corresponds to the engine/service associated with the label. One suitable mechanism is an adaptation of the TCPA TPM_SEAL and

TPM_UNSEAL mechanisms. Recall that a TPM exports UNSEALed plaintext data out of the TPM only if the current pcr-value of pcr-index in the platform matches the decrypted value of [pcr-index, pcr-value]. For the purposes of this invention, a label is used as the plaintext data, and the pcr-index and pcr-value are values that correspond to the software state of the engine/service associated with the label. Alternatively, a TPM could simply store the labels and PCRs, and be arranged to release them only when the correct software state is detected. Preferably a set of labels and PCRs is preloaded into the TPM before deployment of the platform. This reduces the burden on the user of a platform, since platforms are already supplied with a useful set of identities. Any upgrades or maintenance of these identities is preferably done by an entity, preferably the domain, which is trusted by all third parties that will interact with the platform. To perform an upgrade or do maintenance, the domain preferably determines the platform state that it wishes to correspond to some particular engine/service and broadcasts the same information to all platforms within the domain. Part of the broadcast is preferably the distribution of new or upgraded application/domain software. Another part of the broadcast is preferably the distribution of the corresponding new or upgraded software measurements (PCRs) to TPMs, such information being signed using the domain's private key. Each TPM preferably verifies the new information in the same way, using the same public key preloaded into the TPM at manufacture or initialization.

A label is preferably not confirmed by a platform identity unless the platform's software state is such that the